

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-282141

(43)Date of publication of application : 29.10.1993

(51)Int.Cl.

G06F 9/06

G06F 11/28

(21)Application number : 04-076553

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 31.03.1992

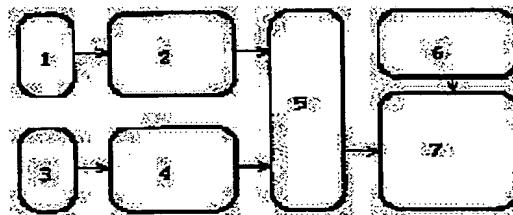
(72)Inventor : YAMAUCHI NOBUYUKI

## (54) DEVICE FOR SUPPORTING EVALUATION OF SOFTWARE DEVELOPING ENVIRONMENT

## (57)Abstract:

**PURPOSE:** To provide a software developing environment evaluation supporting device capable of easily changing data for a developing process or a developing environment.

**CONSTITUTION:** A process model is inputted from a process model input part 1 and recorded in a process model storing part 2. An environment model is inputted from an environment model input part 3. An environment model storing part 4 is constituted differently from the storing part 2. Both the models have mutually corresponding hierarchical structure and dependently connect slave processes describing the contents of respective processes more detailedly. An evaluation engine 5 calculates which tool to be simply used for the developing process is included in the developing environment and evaluates suitability. In this case, recursive algorithm is used. An evaluation result is outputted from a tool existence output part 7 for a certain developing process.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C) 1998,2000 Japanese Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-282141

(43)公開日 平成5年(1993)10月29日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 9/06  
11/28

識別記号 庁内整理番号

4 4 0 U 8944-5B  
3 3 0 A 9290-5B

F I

技術表示箇所

(21)出願番号

特願平4-76553

(22)出願日

平成4年(1992)3月31日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 山内 信之

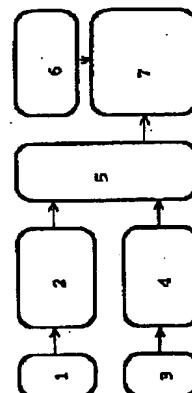
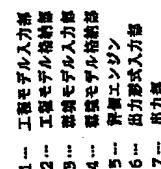
神奈川県川崎市幸区柳町70番地 株式会社  
東芝柳町工場内

(74)代理人 弁理士 木内 光春

(54)【発明の名称】 ソフトウェア開発環境評価支援装置

(57)【要約】

【目的】 開発工程または開発環境のデータの変更が容易なソフトウェア開発環境評価支援装置を提供する。  
【構成】 工程モデル入力部1から工程モデルを入力し、工程モデル格納部2に記録する。環境モデル入力部3から環境モデルを入力する。環境モデル格納部4は、工程モデル格納部2とは異別に構成する。両モデルは相互に対応する階層構造とし、各工程の内容をさらに詳しく記述する下位工程を従属性に接続する。評価エンジン5で、前記開発環境が前記開発工程に簡易に利用できるツールをいかに持っているかを所定の式で算出し、適合性を評価する。この際、再帰的アルゴリズムを用いる。ある開発工程を対象としたツールの存在の有無出力部7から評価結果を出力する。



1

## 【特許請求の範囲】

【請求項1】組込みソフトウェアの開発工程を所定の形式で階層的にモデル化した工程モデルを記録する工程モデル記録手段と、組込みソフトウェアを開発するための開発環境を、前記工程モデルに対応付けて、所定の形式で階層的にモデル化した環境モデルを記録する環境モデル記録手段とを有し、

前記工程モデル記録手段と環境モデル記録手段は、異別に構成され、前記開発環境の前記開発工程への適合性を評価する評価手段を有することを特徴とするソフトウェア開発環境評価支援装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、自動的かつ定量的な評価によって、ソフトウェア開発環境の評価を支援するソフトウェア開発環境評価支援装置の改良に関するもので、特に、組込みソフトウェアの開発環境の評価に適したものに係る。

## 【0002】

【従来の技術】従来から、エア・コンディショナーや自動車のエンジンなどの各種機械に組込み、組込まれた機械を制御する組込みソフトウェアが知られている。近年、これらの機械類の制御には、高機能で複雑な32ビットMPU(マイクロプロセッサ・ユニット)のような高位のマイクロプロセッサを用いる場合が増加している。このため、このようなマイクロプロセッサ上で使用される組込みソフトウェアも高機能化、複雑化する傾向が一般化しつつある。

【0003】また、組込みソフトウェアが使用されるのは、その性質上、エア・コンディショナーや自動車のエンジンなどの機械類であるのに対して、ソフトウェアの開発に最も適るのは、開発用コンピュータである。すなわち、ソフトウェアの開発には、一般に、プログラムを編集するテキスト・エディタ、エディタで作成されたソース・コードを機械語のオブジェクトに翻訳するコンパイラ、オブジェクトの誤りを解析するデバッガなどを用いるが、これらの使用には、ディスクドライブ、モニタ画面、キーボードなどの機器を備えたコンピュータが必要だからである。

【0004】したがって、組込みソフトウェアの開発においては、ソフトウェアを開発する開発ホストマシンと、開発されたソフトウェアを実際に使用するターゲットマシンが別個のものとなる分散化の傾向が生じている。特に、ターゲットマシンとはまったく別個の開発ホストマシン上で、ターゲットマシンに制御される機械の発生信号や動作を疑似的に再現する各種エミュレータの提供は、この分散化の傾向に拍車をかけた。

【0005】以上のように、そもそも高機能化・複雑化

2

している組込みソフトウェアを、さらに、ターゲットマシンとは異なった開発ホストマシン上で開発している状況では、その開発工程も極めて複雑かつ大規模なものとならざるを得ない。

【0006】このような複雑な組込みソフトウェア開発の効果的な実行は、全面的に開発者自身の技術のみに頼ってはもとより、各種のハードウェア資源を十分に活かすことなしには、到底不可能である。特に、人間の能力のみに全面的に頼り、またはハードウェア資源を十分活用せずに開発を行えば、複雑な組込みソフトウェアのシステム各部の整合性を維持することは困難で、また、プログラム上の誤り(バグ)が発生した場合の保守性に劣り、さらに、非能率的な手作業は開発効率を著しく低下させてしまう。

【0007】この組込みソフトウェアの効果的開発に不可欠なのが、開発環境である。ここで開発環境とは、ソフトウェアの開発に適した環境を提供するハードウェアもしくはソフトウェアまたはこれらを組み合わせたシステムであり、いわゆる開発支援ツールを含む概念である。

【0008】開発環境は通常、開発ホストマシン上に、またはそれに関連して提供され、テキスト・エディタ、コンパイラ、デバッガなどの基本的開発支援ツールを備えている。また、前記エミュレータのほか、場合によつては、プログラムから呼び出すための各種汎用サブルーチン・ライブラリ群や、プログラム構造を組込みソフトウェアの用途に合わせて最適化するオブディマイザなど、特に有用なツールを備えているものも多い。開発環境は、これらの各種ツールによって、組込みソフトウェア開発を支援するもので、近年の複雑化した組込みソフトウェア開発は、この開発環境なしには、もはや考えられない程度に高度化、専門化している。

【0009】しかし、開発環境に関して最も重要なことは、開発しようとする組込みソフトウェアに最適な開発環境を選択し、採用することである。そして、最近では、多くのメーカーからそれぞれ特徴のある組込みソフトウェア開発環境が提供されているので、ユーザである開発者には、その中から自己の開発する組込みソフトウェアに適当と思われる開発環境を選択する可能性が与えられている。

【0010】しかし、かかる組込みソフトウェア開発環境は一般に高価であり、ユーザは必要な開発工程に適した開発環境を慎重に選択しなければならない。また、この前提として、各開発環境がどの程度必要な開発工程に適しているかを評価する必要がある。この評価については、開発工程のデータと開発環境のデータから、開発環境の、特定開発工程への適合性を自動的、定量的に行う評価装置も知られている。

【0011】

【発明が解決しようとする課題】ここで、組込みソフト

ウェアの開発には、ターゲットマシンと開発ホストマシンが異なるという、分散化の特性があるため、その開発環境評価においては、開発工程または開発環境のうち一方のデータを変更して評価を繰り返すという特殊な必要が生ずる。しかし、従来の評価装置はこの点を考慮しておらず、組込みソフトウェアの分散化した開発環境の評価に適したものではなかった。このため、従来の評価装置は、次のような様々な欠点を有していた。

【0012】(1) 従来の評価装置においては、開発工程に関するデータと、開発環境に関するデータが独立しておらず、一体に記録されていた。このため、特に、評価アルゴリズムを実行する装置とデータ変更を行う装置が異別の、いわゆる分散環境の場合、一方のデータを変更するためには、他方のデータまでも一緒にデータ変更を行う装置に転送しなければならないという問題点があった。

【0013】(2) また、従来は、開発環境に関するデータと開発工程に関するデータの形式が異なっており、両データ間の整合性が欠如していた。このため、特に、膨大なデータが存在する場合において、一方の変更にともなって他方のデータの対応する部分を変更するためには、これを発見しおよび修正するための手順が複雑化するという問題点があった。

【0014】(3) さらに、従来は、データの幅広い変更を念頭に置いておらず、データおよびアルゴリズムの双方が、特定の開発工程または開発環境のために最適化されていた。特に、データが必ずしも階層化されていなかっただため、データ変更にアルゴリズム自身が対応する再帰的アルゴリズムが採用できず、一定以上のデータの変更には、特別なアルゴリズムの付加が必要であった。このような評価装置自体の変更は、評価自体の低速化を生じるのみならず、評価装置の保守負担をも増大させるという問題点があった。

【0015】以上の結果として、従来の評価装置において組込みソフトウェア開発環境の評価を行おうすると、データの変更が煩雑で、特に、複数の開発工程または開発環境についての評価結果の対比が困難という欠点があった。

【0016】このため、従来の組み込みソフトウェア開発環境の選定は、ほとんどの場合、各製品それぞれのカタログに記載された諸元（スペック）にたよるという原始的なやり方で行わざるを得ず、このための労力は甚大なもので、組込みソフトウェア開発の無視できない負担となっていた。

【0017】本発明は上記の従来技術の問題点を解決するために提案されたもので、その目的は、開発工程または開発環境のデータの変更が容易なソフトウェア開発環境評価支援装置を提供することにある。

【0018】

【課題を解決するための手段】上記の目的を達するた

め、本発明のソフトウェア開発環境評価支援装置は、組込みソフトウェアの開発工程を所定の形式で階層的にモデル化した工程モデルを記録する工程モデル記録手段と、組込みソフトウェアを開発するための開発環境を、前記工程モデルに対応付けて、所定の形式で階層的にモデル化した環境モデルを記録する環境モデル記録手段とを有し、前記工程モデル記録手段と環境モデル記録手段は、異別に構成され、前記開発環境の前記開発工程への適合性を評価する評価手段を有することを特徴とする。

【0019】

【作用】以上の構成を有する本発明のソフトウェア開発環境評価支援装置は、次のような作用を有する。すなはち、本発明においては、工程モデルと環境モデルが異別に記録されているので、一方のみをデータ変更を行う他の装置に転送でき、データの変更の操作が容易になる。

【0020】また、工程モデルと環境モデルが対応付けて記録されているので、データの整合性が保たれ、一方の変更に対応する他方の変更点の発見と修正が容易になる。さらに、両モデルが階層化されているため、汎用性の高い再帰的アルゴリズムが使用でき、データの変更があってもアルゴリズムをそのまま使用することができる。

【0021】

【実施例】以下、本発明の一実施例について、図面1から7に基づいて説明する。

【0022】(1) 実施例の構成

本実施例のソフトウェア開発環境評価支援装置は、図1に示すように、工程モデル入力部1と、工程モデル格納部2と、環境モデル入力部3と、環境モデル格納部4と、評価エンジン5と、出力形式入力部6と、出力部7とから構成されている。

【0023】このうち、工程モデル入力部1は、組込みソフトウェアの開発工程を所定の形式で階層的にモデル化した工程モデルを入力する入力手段である。工程モデル格納部2は、この工程モデルを記録する工程モデル記録手段である。環境モデル入力部3は、組込みソフトウェアを開発するための開発環境を、前記工程モデルに対応付けて所定の形式で階層的にモデル化した環境モデルを入力する第2の入力手段である。

【0024】環境モデル格納部4は、この環境モデルを記録する環境モデル記録手段であって、工程モデル格納部2とは異別に構成されている。評価エンジン5は、前記開発環境の前記開発工程への適合性を評価する評価手段である。出力形式入力部6は前記評価の評価結果の出力形式を入力する部分であり、出力部7は前記出力形式で評価結果を出力する部分である。

【0025】(2) 実施例の作用

以上の構成を有する本実施例のソフトウェア開発環境評価支援装置を用いた開発環境の評価は、図2に示すように、開発工程モデルの入力、開発環境モデルの入力、出

力形式の入力、評価結果の出力の順で行われる。

【0026】ここでは、一つのホストコンピュータと複数のSBC (Single Board Computer) を用いた一連の開発工程に対して、この各要素となる工程に対応する各ツールを有する開発環境がどの程度適合するかを評価する場合を例にとって説明する。なお、図3は、本実施例の評価に係る開発工程の各要素と開発環境の各ツールの対応例である。

【0027】[a. 工程モデルと環境モデルの入力] ユーザはまず、工程モデル入力部1を使用して、工程モデルを対話的に入力する。工程モデルは、工程モデル格納部2に格納される。工程モデルの各工程のデータは、工程番号、工程内容、重み、下位階層へのポインタ、下位階層の工程数から構成されている。ここで、図4に工程モデルの一例を示す。

【0028】工程モデルは階層構造を有し、各工程の内容をさらに詳しく記述する下位工程を従属的に接続することができる。接続には、上位工程のデータのうち、下位階層へのポインタに下位工程の工程番号を記録すればよく、階層化は何段になっていてもよい。なお、ポインタは、下位の階層が存在しない場合には不存在を意味するNULL値を格納する。

【0029】重みは、その工程の重要度で、評価の数値がこれに基づいて算出される。重みは0.0～1.0までの範囲を持つ数値であり、それぞれの工程が全体のうちで占める重要度の割合で示される。重みは、ユーザが定めた適当な値であり、例えば、工程の所要時間が、スケジュール上与えられている時間の全体に占める割合などである。したがって、重みは各開発工程に依存し、同じ工程内容でも開発しようとする組込みソフトウェアによっては異なる数値になる場合もありうる。なお、この重み付けはつきの式が成立つように定められる。

【0030】

【数1】

$$\sum_{i=1}^M \text{重み}_i = 1$$

上式で、Mは全工程数であり、重み $i$ は $i$ 番目の工程の重みを意味している。次にユーザは、評価したい開発環境について、環境モデルを環境入力部3を使用して対話的に入力する。環境モデルは環境モデル格納部4に格納される。環境モデルを構成する各データは、ある工程に関する環境の要素を意味する工程番号や、その工程を対象としたツールの存在の有無、作業効率の程度が含まれる。ここで作業効率の程度とは、その要素の所要時間を示すものであり、たとえば装置への入力コマンド数、あるいは打鍵数などで表す。ここでは装置への入力コマンド数を作業効率の程度を表す単位として用いる。

【0031】また、この環境モデルは、工程モデルと同

様に階層構造を有し、下位の階層の環境モデルとポインタで接続されている。なお、環境モデルの各要素は同一の工程番号によって工程モデルと関連づけられていて、それぞれ対応する工程内容のデータはすぐに引き出せるようになっている。ここで、環境モデルの一例を図5に、また、工程モデルと環境モデルの対応付けの一例を図6に示す。

【0032】[b. 評価] 工程モデル及び環境モデルが入力された後、評価エンジンがこれらのモデルに基づいて、評価を行う。このとき、環境モデルの各要素ごと評価を現す評価値の算出には、評価関数を用いるが、その一例としては次のような式が考えられる。

【0033】

$$\text{一工程の評価値} = \text{重み}/\text{入力コマンド数}$$

この式は、所要時間の長い工程を少ない入力コマンド数で処理できるほど、その工程の評価値が高くなることを意味する。

【0034】また、評価エンジンの動作のアルゴリズムを図7に示す。なお、SUBモジュールは自分自身からの複数階層的再帰呼出が可能なので、便宜上、親であるメインルーチンから直接呼び出されているSUBモジュールを子供、子供から呼び出されたSUBモジュールを孫と呼ぶ。なお、SUBモジュールの呼び出しに最低限必要な引数は、工程数Nとポインタの初期値 $p_t r$ である。そのほか評価関数には、この例では入力コマンド数、重みが必要となる。またこれらの引数はモジュールごとにすべてローカル変数であり、外部からは影響を受けない。但し、変数SUMは、評価値を積算するためのグローバル変数である。

【0035】なお、親からポインタ $p_t r$ と工程数Nとを引き渡されて下位階層に下ろされた子供、孫などのSUBモジュールは、一般に、変数 $i$ をカウンタに用いて、 $p_t r$ の示すデータからN個の工程データの評価値を順にSUMに加算し、親の元へ帰る。

【0036】すなわち、子供は各工程データに下位階層へのポインタがあれば、自分の子供にそのポインタと工程数を引き渡して下位層を処理させ、なければ対応する環境要素のツールの有無を調べ、ツールがあれば、次へ進む前に、評価関数で算出した評価値をグローバル変数SUMに加算する。

【0037】このようなSUBモジュールを最初に呼び出すのは、評価エンジンのメインルーチンで、引数としては、 $p_t r$ に1、第1階層工程数がNに与えられる。

【0038】子供は、工程データの下位階層へのポインタがあれば、その工程はポインタが指す第2階層の工程に展開されているので、展開された第2階層のそれぞれの工程の評価値の和を第1階層の評価値とする。第2階層の工程も下位工程を持つときは、子供はさらに孫を呼び出し、自分と同様にポインタを追跡させる。このポインタの追跡は、下位階層のポインタの値がなくなるま

7  
で、原則として無限代に続けられる。

【0039】例えば、第1階層の「デバッグ」工程が第2階層へのポインタを持つ場合は、「デバッグ」工程の評価は、これに対応する第2階層の各工程の評価値の和となる。但し、この場合の第2階層に、下位階層へのポインタがまったく存在しないときは第3階層は存在しないので、ポインタの追跡は第2階層までで終了する。\*

$$\text{全体の評価値} = \sum_{i=1}^N (\text{重み } i / \text{作業効率の程度})$$

この評価値は0.0~1.0の間で定量的に算出される。そして、すべての工程に開発ツールが用意され、しかも各ツールのコマンド入力数が1の場合は、評価値は最高の1.0となる。一方、いかなる工程にもまったくツールが存在しない場合は、評価値は最悪の0.0となる。

【0042】得られた結果は出力部7に渡され、出力部7はユーザが出力形式入力部6から入力した形式にしたがって、表示を行う。

【0043】[c. データの変更] データの変更は工程モデルと環境モデルのいずれか一方または双方について行うことができる。この場合、データ変更を行う装置が各モデル格納部と別体に構成されている場合は、変更する一方のみをその装置に転送して編集することができる。

【0044】また、一方の変更にともなって他方の変更が必要な場合は、双方に付されたポインタ番号によって容易に対応部分を発見し、修正することができる。

【0045】さらに、本実施例では再帰的アルゴリズムを用いている。この再帰的アルゴリズムはその性質上、アルゴリズム自体がデータ構造に追従する。したがって、下位階層へのポインタの増設または廃止のみで、下位階層への再帰的呼び出し動作を付加または抑制することができる。したがって、1または2以上のデータ単位を付加または削除した場合も、これに相当するポインタまたは工程数の変更のみで、その部分を評価の対象に付加または削除することができる。このため、データ変更に伴うアルゴリズムの変更は不要である。

【0046】(3) 実施例の効果

本実施例によれば、開発工程に関するデータと、開発環境に関するデータが独立して記録されている。このため、特に、評価アルゴリズムを実行する装置とデータ変更を行う装置が異別の、いわゆる分散環境の場合、一方のデータを変更するために、他方のデータまでもデータ変更を行う装置に転送する必要はない。

【0047】また、開発環境の関するデータと開発工程に関するデータの形式が同一であって、両データ間に整合性が保たれている。このため、特に、膨大なデータが存在する場合、一方の変更にともなう他方のデータの対

\* 【0040】以上を第1階層の最後の工程まで行い、その結果得られる第1階層の各工程の評価値を足すことによって、全体の開発環境の評価値を算出する。したがって、開発環境全体の評価値は前記の評価関数を含む次の式で定義される。

【0041】

【数2】

応する部分の変更においてもその発見と修正の手順の複雑化を回避することができる。

【0048】さらに、データの幅広い変更を念頭に置き、両モデルが階層化されているため柔軟性の高い再帰的アルゴリズムが使用でき、データの変更があっても、内蔵アルゴリズムをそのまま使用することができる。このため、データの幅広い変更の場合も、特別なアルゴリズムの付加は必要なく、評価装置自体の変更は不要なので、従来よりも評価自体を高速化できるのみならず、評価装置の保守負担をも軽減させることができる。

【0049】以上のように、本実施例によれば、組込みソフトウェア開発環境の評価を行う場合のデータ変更が容易で、特に、複数の開発工程または開発環境についての評価結果の対比が容易になる。

【0050】また、両モデルは、階層的に記述可能であるから、ユーザは、最初に概略を入力し、後に詳細を入力するという理解容易な構造的モデル記述が可能である。特に、本実施例における評価アルゴリズムは、再帰的に定義されているので、アルゴリズムの記述が簡潔になる。

【0051】

【発明の効果】以上のように、本発明によれば、データの変更が容易なソフトウェア開発環境評価支援装置を提供することができる。

【図面の簡単な説明】

【図1】本発明の実施例の構成図。

【図2】同実施例における評価作業の流れ図。

【図3】同実施例の評価に係る開発工程の各要素と開発環境の各ツールの対応例。

【図4】同実施例の工程モデルの一例。

【図5】同実施例の環境モデルの一例。

【図6】同実施例における工程モデルと環境モデルの対応付けの一例。

【図7】同実施例の評価アルゴリズムの流れ図。

【符号の説明】

1… 工程モデル入力部

2… 工程モデル格納部

3… 環境モデル入力部

4… 環境モデル格納部

(6)

特開平5-282141

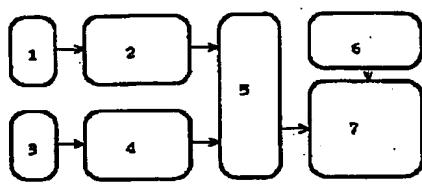
9

10

5… 評価エンジン  
6… 出力形式入力部

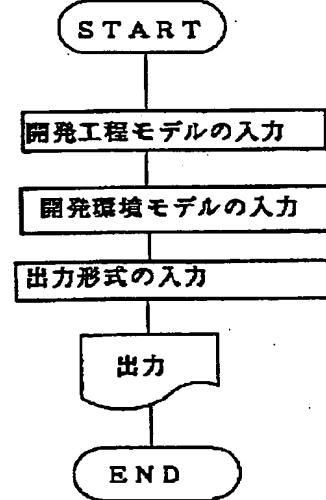
\* 7… 出力部  
\*

【図1】



1… 工程モデル入力部  
2… 工程モデル検証部  
3… 理論モデル入力部  
4… 理論モデル検証部  
5… 評価エンジン  
6… 出力形式入力部  
7… 出力部

【図2】



【図4】

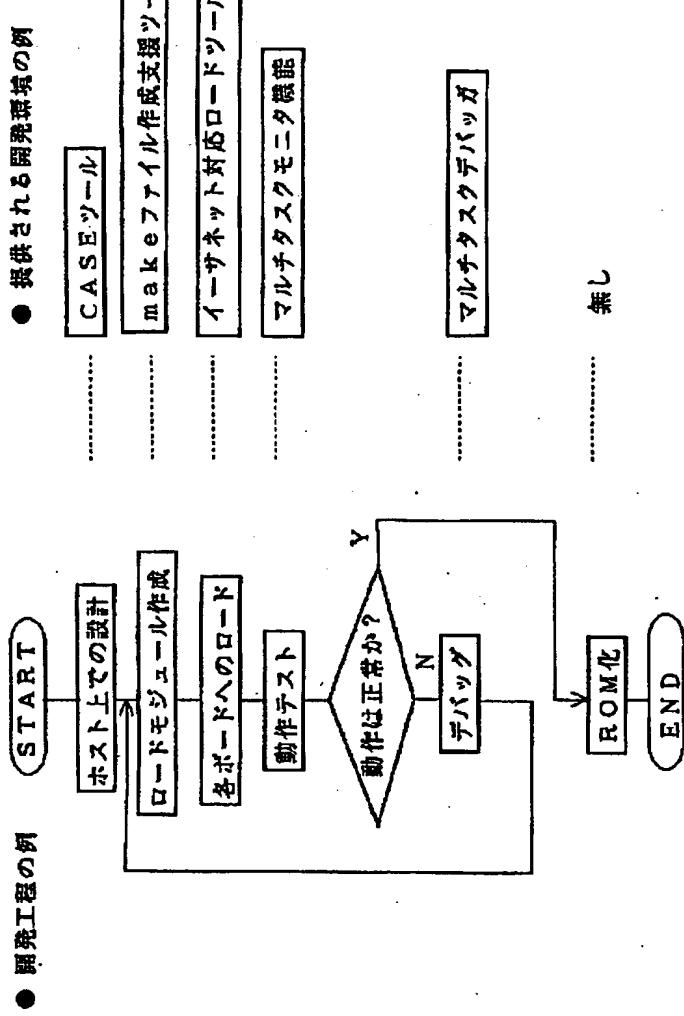
第1階層

第1工程工程数	
0	1 (工程番号) ホスト上での設計 (工程内容)
3	(重み)
0	(下位階層へのポイント)
11	(下位階層の工程数)
⋮	
5	デバッグ
7	
90	
5	

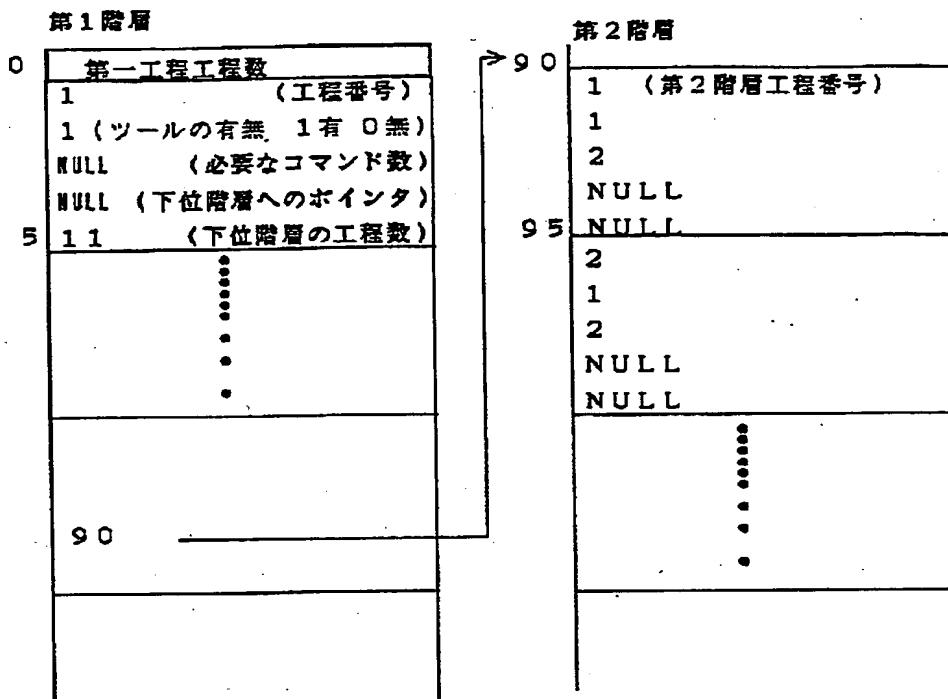
第2階層

⇒ 90	1 (第2階層工程番号) ボード1上でデバッガ起動
95	1 NULL NULL
	2 ボード2上でデバッガ起動
	3 NULL NULL
	⋮
5	

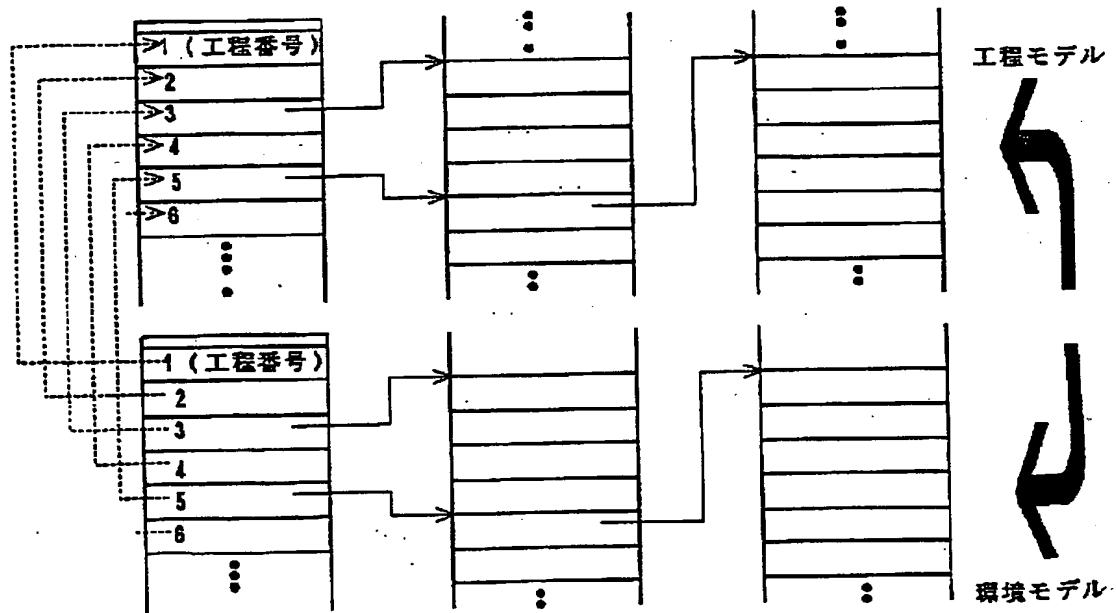
[図3]



[図5]



【図6】



【図7】

